REMARKS

The claims are claims 1 to 22.

The application has been further amended at many locations to correct minor errors and to present uniform language throughout. The amendments include deletion of the reference numbers in the ABSTRACT.

The amendments include change of the TITLE OF THE INVENTION to "PROCESSOR WITH CONDITIONAL INSTRUCTION EXECUTION BASED UPON STATE OF CORRESPONDING ANNUL BIT OF ANNUL CODE." This new TITLE OF THE INVENTION is believed properly reflective of the subject matter of this application.

Claims 1, 3, 5 to 9, 16 to 18 and 20 are amended. New claim 22 is added.

Claims 1 to 21 were rejected under 35 U.S.C. 103(a) as made obvious by the combination of Ohnishi et al U.S. Patent No. 4,566,062 and Babaian et al U.S. Patent No. 5,958,048.

Claim 1 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 1 recites "storing an annul code having an annul bit corresponding to each instruction of a group of instructions in the pipeline." The dummy cycle instruction disclosed in Ohnishi et al does not have an annul bit corresponding to each instruction in a group of instructions. Accordingly, claim 1 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 1 recites further subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 1 recites "circuitry for preventing one or more selected instructions in the group from altering the architected state in response to the corresponding annul bit of the annul code." The OFFICE ACTION cites the ABSTRACT; column 1, lines 6 to 10, lines 31 to 40 and lines 53 to 57; column 2, lines 31 to 40 and 55 to 64; and column

3, lines 10 to 19 and 57 to 67 of Ohnishi et al as making obvious this limitation. The Applicant respectfully submits that this argument is incorrect. Ohnishi et al teaches a dummy instruction with inserts a controllable number of delay cycles into the instruction stream. Ohnishi et al states at the ABSTRACT lines 4 to 6:

> "Dummy cycles for a number of cycles having a processing time equal to the desired waiting time are generated by a dummy instruction."

This indicates that the dummy instruction is used to insert waiting time and not to annul the operation of an instruction. Ohnishi et al states at column 1, lines 6 to 10:

> "This invention relates to a timing protection system for various control timings in a data processor, and more specifically to the realization of the timing function by the use of a dummy cycle executed by an instruction, particularly a micro-instruction."

This indicates that the "dummy cycle executed by an instruction" is used for various control timings. Ohnishi et al states at column 1, lines 31 to 40:

> "In order to attain such an object in a data processor, an instruction for executing a dummy cycle which is invalid for processing is inserted at the desired position in the program. In the above example, the instruction is inserted between the first and second micro-instructions for which timing protection is necessary. Within the instruction, the number of repetitions of the dummy cycle is designated and the protection time is a function of the product of the number of repetitions of the dummy cycle and the cycle time of the processor."

This indicates the dummy cycle instruction "is inserted at the desired position in the program" for timing protection. Ohnishi et al states at column 1, lines 53 to 57:

> "This invention provides the desired timing protection during execution by inserting an instruction designating a dummy cycle at the position in the program requiring timing protection. The number of dummy cycles are controlled by the instruction."

This indicates that the dummy cycles are implemented by placing an instruction at the position requiring timing protection. Ohnishi et al states at column 2, lines 31 to 40:

> "FIG. 2 is a block diagram of a hardware structure used for execution of a dummy cycle in this embodiment hereunder. This structure is connected to the pipeline control structure of FIG. 1. Of the micro-instruction words read from the control memory CS 2, the control fields called instruction TAG control phase-A (ITCA), execution mask EM, phase-B loop control (PBLC), end loop (EL) and iteration counter for phase-B (ITCB) are used in order to realize a dummy cycle. Functions of each field will be described later."

This disclosure of Ohnishi et al fails to teach or suggest that any structure of Figure 2 prevents an instruction "from altering the architected state" as recited in claim 1. Ohnishi et al states at column 2, lines 55 to 64:

> "A dummy cycle in this embodiment executes a loop for counting one by one the values present in the EM field in the ITC register 20. The time until ITC=0 is the protection or reserved time. The contents of the ITCA field in the ITCA register 11 are decoded in the decoder 16 at the end of PHASE-A and a control signal is output so that the value of the EM field is input to the ITC register 20 from the EM register 12. The value of the EM field in the EM register 12 determines the number of loops of the dummy cycle."

This disclosure of Ohnishi et al involves executing a loop of dummy cycles but fails to teach or suggest preventing an instruction "from altering the architected state" as recited in claim 1. Ohnishi et al states at column 3, lines 10 to 19:

> "When the single loop latch 24 is being set and the END LOOP is not ON, the PHASE-B valid flag is set again for each end of PHASE-A by the gates 25 and 28, but it is not reset even for the end of PHASE-B. In other words, the PHASE-B valid flag is kept ON. The output of AND gate 27 is used as a set enable signal to the PHASE-B-TAG register 4 and the contents of the PHASE-B-TAG register 4 are kept constant until this signal is generated when the END LOOP signal appears."

The Applicant respectfully submits that this description of elements of Figure 3 fails to teach preventing an instruction group "from altering the architected state in response to the corresponding annul bit of the annul code." Ohnishi et al states at column 3, lines 57 to 67:

> "As explained above, a time-out is generated for the period (EM+1).times.2.times.(machine cycle) when a a desirable value is present in the EM field and the timing is protected during such period. The method of realizing a dummy cycle is not restricted to precisely the above method and other adequate methods can also be obtained as required. According to this invention, a desired timing protection can be easily provided by inserting a single micro-instruction in an existing micro-program processor's control memory and providing the necessary control circuits."

This portion of Ohnishi et al teaches insertion of dummy cycles for desired timing protection but fails to teach preventing an instruction "from altering the architected state in response to the corresponding annul bit of the annul code." This is particularly true because Ohnishi et al discloses that the instruction following the dummy cycle instruction is inhibited until the number of dummy

cycles completes. Ohnishi et al states at column 3, lines 33 to 35:

> "Therefore, operation of PHASE-B is executed again and the next instruction (NEXT INST.) is inhibitted from moving from PHASE-A to PHASE-B and is nullified."

Ohnishi et al likewise states that the next instruction is prevented from moving in the pipeline at column 3, lines 43 to 45. Ohnishi et al further states at column 3, lines 52 to 56:

> "When the single loop latch becomes OFF, the dummy cycle terminates and the next instruction (NEXT INST.) is allowed to move to PHASE-B from PHASE-A by the output of gate 27, PHASE-B-TAG SET signal."

The Applicants respectfully submit that Ohnishi et al teaches mere delay of completion of the next instruction and not the prevention from altering the architected state as recited in claim 1. Accordingly, claim 1 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 3 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 3 recites "in response to an annul bit being a first state the execution unit...does not execute the corresponding instruction in the given clock cycle" and "in response to an annul bit being a second state different than the first state the execution unit...does execute the corresponding instruction in the given clock cycle." Paragraphs 11a, 11b, 11c and 11d spanning pages 4 and 5 of the OFFICE ACTION cite the same portions of Ohnishi et al already quoted above as making obvious the four limitations of claim 3. The Applicant respectfully submits that none of these portions of Ohnishi et al make obvious execution or non-execution of an instruction dependent upon the bit state of a corresponding bit.

As noted above, Ohnishi et al teaches delay in execution of an instruction and not the "does not execute" limitation of claim 3. Accordingly, claim 3 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 5 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 5 recites "an integer number N of the plurality of execution units are scheduled to execute" on a particular cycle and "wherein the circuitry for coupling the annul bits to respective ones of the plurality of execution units comprises circuitry for coupling only the integer number N of the annul bits to the plurality of execution units which are scheduled to execute in the given clock cycle." The OFFICE ACTION at paragraphs 15b and 15c cite Babaian et al at column 1, lines 51 to 60, column 5, lines 38 to 44 and column 23, lines 1 to 15 as making these limitations obvious. The Applicant respectfully submits that these portions of Babaian et al teach parallel execution of plural instructions in corresponding execution units. However, these portions of Babaian et al fail to make obvious an annul code of annul bits corresponding to individual instructions, nor that "only the integer number N of the annul bits" are coupled "to the plurality of execution units which are scheduled to execute in the given clock cycle." Accordingly, claim 5 is allowable over Ohnishi et al and Babaian et al.

Claim 11 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 11 recites "the annul code is loaded from a memory." The OFFICE ACTION cites column 1, lines 62 to 67 of Ohnishi et al as making this limitation obvious. This portion of Ohnishi et al teaches loading instructions from a memory. However, this portion of Ohnishi et al fails to teach loading the annul code from memory because Ohnishi et al fails to make obvious the recited annul code. Accordingly,

claim 11 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 12 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 12 recites "the annul code is an immediate value in an instruction passing through the pipeline." The OFFICE ACTION cites column 23, lines 4 to 9 of Babaian et al as making this limitation obvious. This portion of Babaian et al states:

> "Long instructions are stored in a memory 211 and an instruction cache (IC) 282 of VLIW processor 200 in packed form as sets of 16- and 32-bit syllables. Particular operations can occupy a part of syllable, a whole syllable or several syllables."

This portion of Babaian et al fails to mention instruction immediate fields and fails to suggest an annul code in such an immediate field. In the absence of any citation to the teaching of an immediate field, this cannot make obvious the specific immediate field recited in claim 12. Accordingly, claim 12 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 17 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 17 recites first and second data registers, "the first annul code is stored in the first data register" and "the second annul code is stored in the second data register." The OFFICE ACTION cites column 1, line 61 to column 2, line 9 of Ohnishi et al as making obvious the recited data registers. However, this portion of Ohnishi et al does not teach data registers but rather pipeline stage registers PHASE-A to PHASE-F which store micro-instructions for the various pipeline stages. Note that the OFFICE ACTION fails to state which of the TAG registers PHASE-A to PHASE-F stores the first annul code and which stores the second annul code recited in claim 17.

Accordingly, claim 17 is allowable over the combination of Ohnishi et al and Babaian et al.

Claim 18 recites subject matter not made obvious by the combination of Ohnishi et al and Babaian et al. Claim 18 recites "a data register" with "the first annul code is stored in a first one-half of the data register" and "the second annul code is stored in a second one-half of the data register different from the first one-half." The OFFICE ACTION states that Ohnishi et al fails to teach the recited storage in half registers. The OFFICE ACTION states that Babaian et al at column 23, lines 4 to 9 teaches this limitation. This portion of Babaian et al, which is quoted above, mentions memory 211 storing instructions and instruction cache 282 but fails to mention a data register. This portion of Babaian et al teaches that particular operations "can occupy a part of syllable, a whole syllable or several syllables." These syllables are presumably a part of instruction memory 211 and instruction cache 282. Babaian et al fails to teach storing two data elements in differing halves of a data register as recited in claim 18. Accordingly, claim 18 is allowable over the combination of Ohnishi et al and Babaian et al.

New claim 22 is allowable over the combination of Ohnishi et al and Babaian et al. New claim 22 recite a manner of forming annul codes not taught or made obvious by the cited references. This method is described in the application at page 12, line 9 to page 14, lines 19 and illustrated in Figure 3a.

The Applicants respectfully submit that al the present claims are allowable for the reasons set forth above. Therefore early reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicants request that the Examiner contact Applicants' attorney at the below listed telephone number and address to facilitate prosecution.

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418

Respectfully submitted,

Robert D. Marshall, Jr.
Reg. No. 28,527